

4902: VIDEOJocs EDUCATIUS: CREACIÓ D'UNA
PLATAFORMA PER A L'ANÀLISI DE LES ESTRATÈGIES
UTILITZADES A UN JOC DE TIPUS TOWER DEFENSE

Memòria del Projecte Final de Carrera
d'Enginyeria en Informàtica realitzat per
Daniel Guzmán Cerón i dirigit per Aura
Hernández Sabaté
Bellaterra,de.....de 2012

La firmant, Aura Hernández Sabaté, professora del Departament de ciències de la computació de la Universitat Autònoma de Barcelona.

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Daniel Guzmán Cerón

Bellaterra, Agost de 2012

Firmat: Aura Hernández Sabaté

Índex

1	Introducció	1
1.1	Marc del Projecte	1
1.2	Estat de l'art	2
1.3	Objectius del projecte	5
1.4	Viabilitat del projecte	5
1.5	Planificació temporal	6
2	Eines	9
2.1	IDE	9
2.2	SDL	9
2.3	PugiXML	10
3	Implementació de l'entorn	11
3.1	Disseny	11
3.2	Mòdul de Renderitzat	12
3.3	Mòdul de Fonts.	14
3.4	Mòdul de gestió d'entrada	16
3.5	Mòdul de gestió del temps	17
3.6	Mòdul de gestió de configuracions XML	18
4	Implementació del joc	19
4.1	Vista global	19
4.2	Lògica i Motor d'Estats	22
4.3	Elements Configurables	25

4.3.1	Nivell	25
4.3.2	Camins	26
4.3.3	Enemics	28
4.3.4	Torres	30
4.4	Estadístiques	33
5	Conclusions	37
5.1	Compliment dels Objectius	37
5.2	Ampliacions o millores	38
5.3	Valoracions d'un professor	39
	Bibliografia	43

Capítol 1

Introducció

1.1 Marc del Projecte

La resolució de problemes és un camp estretament lligat a la pràctica matemàtica i considerat de vital importància en l'aprenentatge de les matemàtiques. Els estudis realitzats en les darreres dècades són realment nombrosos (Lester, 1994; Schoenfeld, 2007), però aquesta abundància de literatura no ha suposat un increment equivalent en l'ús de la resolució de problemes a les aules de secundària.

Engle i Conant (2002) han realitzat una abstracció de les característiques que s'han de complir per aconseguir una bona classe de matemàtiques en la que es treballi la construcció del significat (sense-making) per part del alumnes:

- Els alumnes han de ser encoratjats a resoldre problemes.
- Els estudiants trindran autoritat per triar la direcció en la que es resolen aquests problemes.
- El treball dels alumnes serà avaluat per altres membres del grup i s'haurà d'ajustar a les normes.
- Els alumnes tindran els recursos necessaris per portar a terme tot el que se'ls demana.

Un dels factors fonamentals que afecten la resolució d'un problema per part dels estudiants és el context inherent a aquest. Segons Van Den Heuvel-Panhuizen (2005), introduir un context real als problemes pot augmentar la seva accessibilitat i suggerir estratègies als alumnes. Al mateix temps es considera que les matemàtiques informals són dominants en la resolució de problemes en la vida quotidiana i en el món laboral i, en canvi, les matemàtiques formals són predominants a l'aula. Lave (1998) va comprovar que alguns alumnes que fracassen en les situacions matemàtiques escolars poden ser molt competents en activitats de la vida quotidiana en les que cal utilitzar els mateixos continguts matemàtics. S'ha observat que, en situacions de la vida real, les persones es senten implicades i utilitzen estratègies matemàtiques pròpies per resoldre els problemes que poden ser molt diferents de les que s'ensenyen a l'escola.

Els videojocs avui en dia formen part de la vida quotidiana dels. Es per això que aquest projecte intenta acostar la resolució i avaluació d'estratègies diferents als alumnes de les classes de matemàtiques mitjançant un joc del tipus 'Tower Defense'.

1.2 Estat de l'art

Els joc de defensa de torres 'Tower Defense' són jocs d'estratègia en temps real en els que la presa de decisions és vital per aconseguir l'èxit en els objectius per superar el joc. Les estratègies d'èxit en el joc es basen en una bona gestió dels recursos (presa de decisions), la distribució de les torres al mapa (geometria) i la comprensió del sistema de dany que sofreixen els enemics (relacions numèriques) amb el que aquest tipus de joc combina diferents elements inclosos en els currículums de matemàtiques per a l'educació secundària i es poden utilitzar com a context per introduir la resolució de problemes.

L'objectiu és intentar aturar els enemics abans que arribin al final d'un recorre-

gut per un mapa mitjançant la construcció de torres que els dispararan quan passin per dins el seu radi d'acció. Els enemics i les torres tenen normalment habilitats diferents, costos de construcció i operació diferents. Quan un enemic és derrotat, el jugador aconsegueix diners o punts o recursos que pot invertir en actualitzar les torres que ja té o construir-ne de noves.

El posicionament i la selecció del tipus de torre és la base d'aquests jocs d'estratègia. És aquí on recauen les decisions del jugador per maximitzar el mal als enemics.

El primer joc que es considera un 'Tower Defense' és el [figura 1.1] "Rampart"(1990 Atari) tot i que és força diferent als videojocs moderns d'aquest gènere ja que alhora de disparar les torres s'han d'apuntar manualment, en canvi avui en dia es fa de manera automàtica sense interacció del jugador.



Figura 1.1: Imatge de Rampart Atari Games, 1990

Sobre l'any 2000 van sorgir nombrosos mapes personalitzats per a jocs com l'Starcraft i Warcraft III que van ser els pioners en quant a jocs de torres moderns.

A partir de llavors hi ha hagut una explosió de jocs d'aquest gènere, sobretot gràcies a jocs jugables en el navegador web utilitzant Adobe Flash, el que va fer augmentar considerablement la popularitat del gènere, degut a la simplicitat de la seva producció, com podria ser el [figura 1.2] Vector Tower Defense (Candystand, 2007). També han demostrat la seva popularitat pel fet d'aparèixer en diversos dispositius mòbils amb gran èxit de vendes, jocs com podrien ser el Fieldrunners (Subatomic Studios, 2008) o Plants vs Zombies (PopCap, 2009).

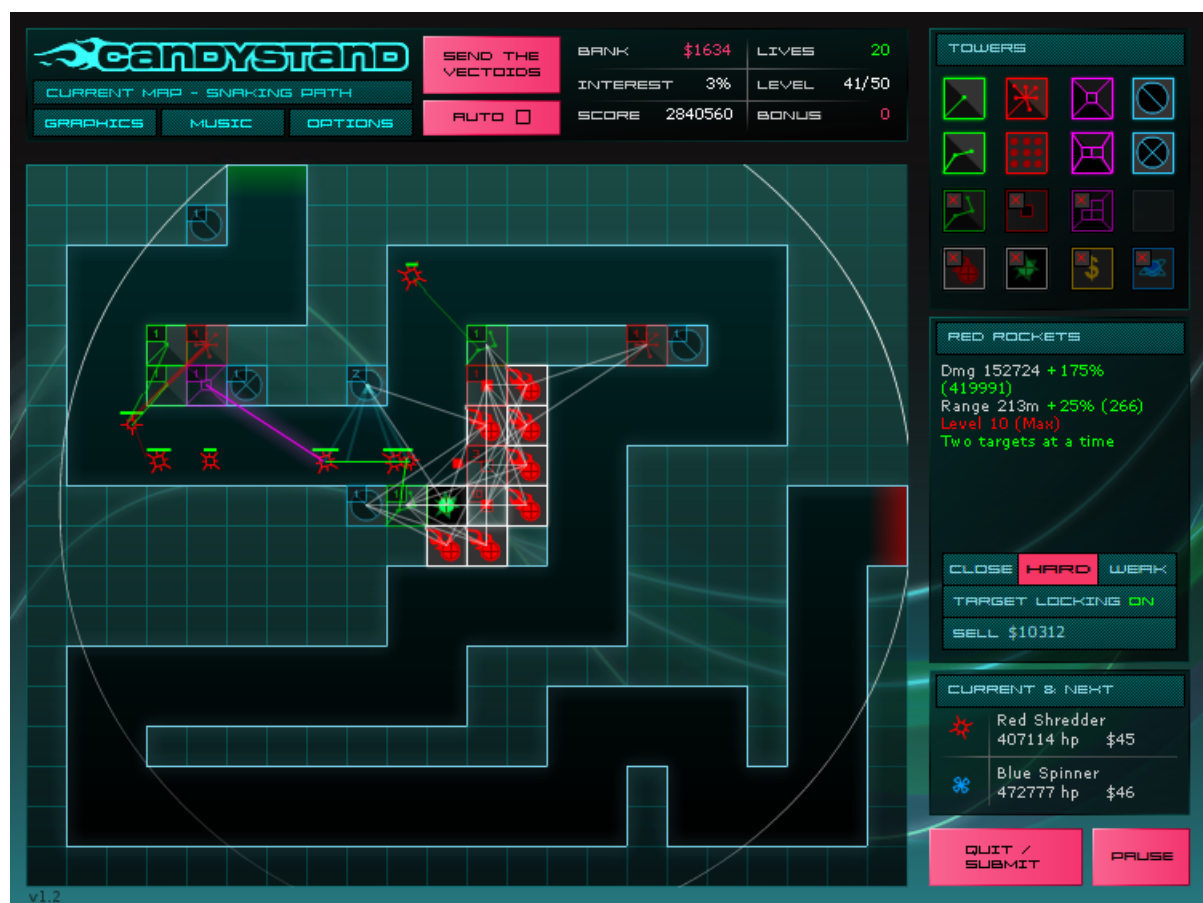


Figura 1.2: Imatge de Vector Tower Defense (Candystand, 2007)

1.3 Objectius del projecte

Amb aquest projecte es pretén dissenyar un software que inclogui un videojoc del tipus defensa de torres amb un nivell adequat de complexitat per a ser utilitzat a les aules de matemàtiques de secundària. Aquest software també ha d'incloure eines que permetin a l'alumne avaluar la seva pròpia actuació per poder generar millors estratègies de joc.

Els requisits que hauria de complir el software desenvolupat són els següents:

- Que estigui dotat d'un entorn gràfic senzill que inclogui els elements principals en un joc de tipus de defensa de torres.
- Que el joc estigui dissenyat de forma neutra sense una ambientació que pugui provocar una discriminació entre els alumnes.
- Que els elements que determinen el nivell de dificultat del joc siguin configurables pel professorat que l'ha de fer servir (camins, enemics, torretes..) per poder-lo adequar a diferents activitats i nivells educatius.
- Que contingui elements que permetin a l'usuari, en aquest cas l'alumne, avaluar el grau de desenvolupament del seu joc.

1.4 Viabilitat del projecte

Aquests jocs de defensa de torres es poden realitzar tant en 2D com en 3D. El software que s'ha realitzat és en 2D ja que l'aspecte visual no és el més important en el projecte sinó l'aspecte funcional del joc, unit també a que les característiques tècniques dels PC que hi ha a les escoles de secundària probablement no són suficients per a un joc en 3D.

Davant la diversitat de sistemes operatius que podem trobar als centres educatius, s'ha optat per fer servir com a base la llibreria de baix nivell (SDL - Simple

DirectMedia Layer) en la seva versió 2.0 que s'ocupa de gestionar l'entrada per teclat, per ratolí, el hardware 3D via OpenGL/Direct3D, gestió d'events i el control del gestor de finestres de diversos sistemes operatius de manera unificada.

El llenguatge que s'ha utilitzat és el C/C++ i el compil·lador GCC utilitzant l'IDE Eclipse. D'aquesta manera tot i que s'ha programat íntegrament en Windows, ens assegurem que a l'hora de necessitar una versió per una altra plataforma com pot ser GNU/Linux o MAC OSX sigui un procés de conversió trivial.

Ja que el joc ha de ser configurable pel professorat, s'ha utilitzat la llibreria pugiXML per a extreure la informació dels diversos fitxers XML que contenen la informació de tots els elements del joc.

1.5 Planificació temporal

La primera part del projecte consisteix en realitzar un anàlisi de requeriments dels elements bàsics d'un joc de tipus defensa de torres. Que inclou:

- Analitzar quins requeriments a nivell baix necessita (renderitzat, control de temps (timers), gestió de fonts, gestió d'events)
- La base del motor d'estats del joc.
- Analitzar quins tipus de torres hi haurà i quins paràmetres seran configurables i quins no.
- Analitzar quins tipus d'enemics hi haurà i quins paràmetres seran configurables i quins no.
- Definir com serà la interfície d'usuari i quines accions podrà realitzar l'usuari.

Seguidament es realitza la tasca de programació d'un petit entorn en SDL que consisteix en:

- Programació d'un gestor d'events que sigui capaç de rebre i atendre les peticions per teclat i ratolí.
- Programació d'un gestor de renderitzat capaç de renderitzar punts, línies, i imatges, de manera òptima.
- Programació d'un gestor de fonts per carregar fonts i juntament amb el gestor de renderitzat poder renderitzar text.
- Programació d'un gestor del temps capaç de gestionar temporitzadors que farem servir com a disparadors a diversos events.

A continuació es realitza la tasca de la programació del joc que consisteix en:

- Programació de la interfície d'usuari, tots els elements visibles del joc, tals com els botons, estadístiques i texts
- Programació de la lògica del joc, és a dir tots els canvis d'estats segons els que succeeixi en el joc o des dels events, així com el moviment d'enemics i actualitzacions a l'estadística
- Programació de la configuració de torres/enemics, la càrrega de les configuracions per XML de les característiques dels enemics/torres
- Programació de la visualització de les estadístiques per cada torre.

Tasques	Inici	Fi
<i>Informe previ</i> - Redacció - Entrega	13/04/12	16/04/12
<i>Anàlisi</i> - Analitzar requeriments. - Analitzar el motor d'estats del joc. - Analitzar quins tipus de torres hi haurà i quins paràmetres seran configurables i quins no. - Analitzar quins tipus d'enemics hi haurà i quins paràmetres seran configurables i quins no. - Definir com serà la interfície d'usuari i quines accions podrà realitzar l'usuari.	26/04/12	20/05/12
<i>Motor/Framework en SDL</i> - Programació d'un gestor d'events. - Programació d'un gestor de renderitzat. - Programació d'un gestor de fonts. - Programació d'un gestor del temps.	21/05/12	17/06/12
<i>Programació del Joc</i> - Programació de la interfície d'usuari. - Programació de la lògica del joc. - Programació de la configuració de torres/enemics. - Programació de les estadístiques.	18/06/12	3/08/12
<i>Elaboració de la memòria</i>	03/08/12	12/09/12

Taula 1.1: Planificació inicial del projecte.

Capítol 2

Eines

2.1 IDE

Per realitzar aquest projecte s'ha utilitzat l'Eclipse CDT. Es tracta d'un entorn de programació i multilenguatge multiplataforma escrit en JAVA. El llenguatge de programació per a implementar el joc és el C/C++ ja que les demés llibreries utilitzades també estan escrites en aquest llenguatge y facilitaràa més la implementació.

2.2 SDL

La llibreria SDL (Simple DirectMedia Layer) és una llibreria multimedia multiplataforma dissenyada per proveïr accés de baix nivell al so, al teclat, al ratolí, al joystick, al Hardware d'acceleració 3D via OpenGL i a un framebuffer de video 2D. Normalment utilitzada per a programes de reproducció de videos MPEG, emuladors i per molts jocs populars tan de codi obert com propietaris com per exemple els jocs Civilization: Call To Power o Unreal Tournament 2004.

SDL es capaç de suportar Linux, Windows, Windows CE, BeOS, MacOS, Mac OS X, FreeBSD, NetBSD, OpenBSD, BSD/OS, Solaris, IRIX, i QNX. I el codi conté suport també per a AmigaOS, Nintendo DS, Dreamcast, Atari, AIX,

OSF/Tru64, RISC OS, SymbianOS, and OS/2, tot i que aquests no estan oficialment suportats. Està escrita en C pero pot funcionar nativament en C++ i està distribuïda sota la llicència GNU LGPL versió 2, fet que fa que es pugui distribuir lliurement en programes comercials si es desitja.

Per al projecte s'ha escollit aquesta llibreria perquè dóna la flexibilitat suficient per fer el joc multiplataforma sense restringir-nos a un sistema operatiu en concret. A més facilita molt la tasca de programació proporcionant una interfície unificada als accesos de baix nivell de tot el hardware necessari per fer un joc.

Concretament, la versió utilitzada de la llibreria és la 2.0, tot i que està en fase BETA ja és suficientment estable com per poder fer-la servir sense cap limitació i actualment és qüestió de mesos que s'alliberi públicament la versió estable 2.0.

2.3 PugiXML

La llibreria pugiXML és una llibreria molt lleugera escrita en C++ per al processament de fitxers XML. Consisteix en una sèrie d'interfícies amb grans capacitats per a interaccionar amb els nodes en un fitxer XML. Conté un parsejador de fitxers XML molt ràpid amb suport Unicode.

És una llibreria extremadament portable i molt fàcilment integrable, totalment de codi obert distribuïda sota la llicència MIT fent-la totalment lliure per utilitzar en qualsevol tipus de projecte sigui de codi obert o propietari.

Totes aquestes característiques han fet que en aquest projecte es faci per llegir els XML de configuració que més endavant s'explicaran.

Capítol 3

Implementació de l'entorn

En aquest capítol s'explicarà el disseny de l'entorn realitzat per a fer funcionar el joc així com els seus mòduls amb la seva configuració i implementació.

3.1 Disseny

Per realitzar el joc primerament és necessari una palataforma que ens gestioni les accions més bàsiques d'un joc. Per exemple, rebre quines tecles està apretant l'usuari, en quina posició està el ratolí, funcions ràpides per renderitzar, controlar temporitzadors que el joc pugui necessitar, etc.

Concretament, per al projecte s'han dissenyat els següents mòduls:

- Mòdul de gestió de renderitzat.
- Mòdul de gestió de fonts.
- Mòdul de gestió d'entrada
- Mòdul de gestió del temps
- Mòdul de gestió de configuracions XML

El mòdul que englobarà tots aquests mòduls per unificar el seu accés serà el 'Core'. Aquest actuarà d'interfície entre el joc i els diferents mòduls de la plataforma.

A la figura [figura 3.1] es pot veure la jerarquia final de mòduls

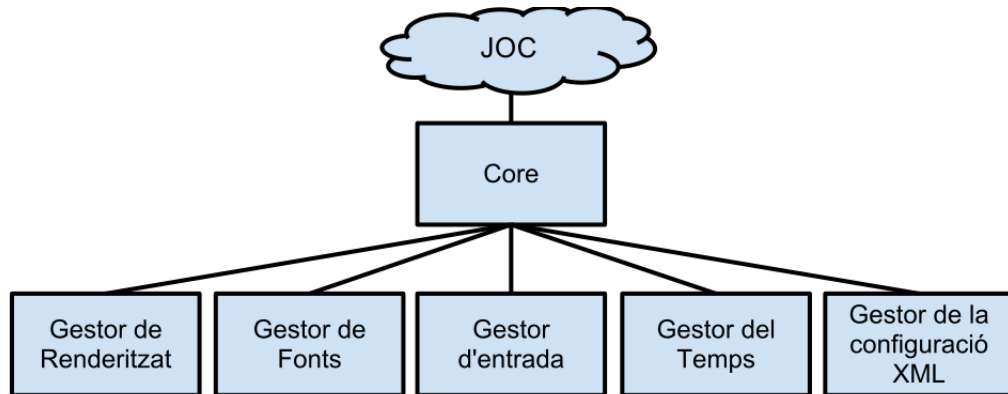


Figura 3.1: Diagrama bàsic de la plataforma

D'aquesta manera, el joc, accedirà al Core per obtenir les funcionalitats més bàsiques que necessita.

3.2 Mòdul de Renderitzat

El mòdul de renderitzat té dos funcionalitats molt importants: la primera és que és l'encarregat de gestionar tot el funcionament de la finestra del joc i de carregar mitjançant el mòdul de configuracions d'XML les característiques d'aquesta. S'ha implementat utilitzant les funcions de baix nivell d'SDL per a renderitzar així com les seves estructures per controlar tots els aspectes de la finestra.

Aquesta configuració es troba al fitxer videooptions.xml i es pot editar per a que el professorat pugui configurar les següents opcions:

- WinWidth - Determinarà l'amplada en píxels de la finestra.

- WinHeight - Determinarà l'altura en píxels de la finestra.
- WinPosX - Determinarà la posició en la coordenada X on la finestra s'obrirà inicialment.
- WinPosY - Determinarà la posició en la coordenada Y on la finestra s'obrirà inicialment.
- Fullscreen - Determinarà si el joc s'executarà en mode de pantalla completa o en mode finestra. 0 - Mode Finestra, 1 - Pantalla completa.
- Renderer - Determinarà quina API volem fer servir per a renderitzar. opengl o direct3d.
- WindowCaption - Determinarà la descripció que sortirà a dalt de la finestra. Per posar-li el nom del joc, per exemple.

La segona funcionalitat del mòdul de renderitzat és pintar a pantalla els diferents elements que el joc necessiti de manera òptima. Aquests elements són els següents:

- Punts - Donada una posició, pintarà el píxel en aquella posició.
- Línies - Donada unes coordenades d'inici i final pintarà una línia. Ens servirà per pintar les línies de la gràfica d'estadístiques i els raigs de les torres.
- Cercles - Donat un punt inicial i un radi generarà una llista de punts que conformaran un cercle. Ens servirà per pintar el radi d'acció de les torres.
- Textures - Donat un identificador de textura i una coordenada de posició, pintarà la textura seleccionada a la posició indicada, això ens servirà a l'hora de pintar les textures dels enemics o les textures de les torres.
- Fonts - Donat un identificador de font, una coordenada de posició i un text, pintarà el text amb la font seleccionada en la posició donada.

Tots aquests elements es guardaran en un mapa indexat per la coordenada Z que indicarà en quina profunditat s'ha de pintar l'element juntament amb la informació sobre el color en que es vol pintar l'element (en cas de que tingui). Al final de cada frame, és a dir quan el joc hagi acabat d'enviar els elements a renderitzar, es recorrerà aquest mapa començant per l'índex Z més llunyà, pintant tots els elements que hi hagi. Un cop finalitzat aquest procés s'esborraran tots els elements del mapa, començarà un nou frame i el joc enviarà els nous elements a renderitzar, així successivament.

Per facilitar que el professorat pugui configurar el joc al seu gust, concretament en l'apartat gràfic, en el fitxer texturesconfig.xml es poden afegir noves textures per què el mòdul de renderitzat les carregui i es puguin fer servir al joc. Per afegir una nova textura simplement afegirem una línia al fitxer texturesconfig.xml amb les següents característiques:

```
<TEXTURA File='XXX' id='XXX' alpha='X' />
```

- File - La ruta relativa l'executable del joc del fitxer a carregar, tenint en compte que només es poden carregar fitxers .bmp
- id - Identificador que se li voldrà donar a aquella textura per més tard utilitzar-la.
- alpha - Ens indica si es vol renderitzar amb alpha o no. 0 - Es renderitzarà la textura sense tenir en compte un color alpha, 1 - Es renderitzarà la textura agafant com a color alpha el píxel [0,0] de la imatge.

3.3 Mòdul de Fonts.

El mòdul de gestió de fonts s'utilitzarà per carregar i parsejar les fonts que hi ha configurades al fitxer fontsconfig.xml. D'aquesta manera aquest mòdul és l'encarregat de seleccionar les lletres del mapa de lletres d'una font quan el joc necessita

pintar un text per pantalla i enviar-li aquesta informació al mòdul de renderitzat per que les pinti adequadament.

S'ha tractat una font com un fitxer d'imatge on hi ha tots els signes del codi ASCII posades en ordre i separades sempre per un mateix offset. Això és així perquè el mòdul de gestió de fonts quan rebi una cadena de caràcters a pintar, miri quin desplaçament ha de recórrer segons el símbol ASCII i el pugui anar a buscar amb aquest mateix desplaçament a la font.

El mòdul de gestió de fonts, compta amb funcions, que alhora de programar, serviran de molta ajuda per situar el text on calgui de manera més senzilla. Les seves funcions d'ajuda a l'hora de pintar són les següents:

- `RenderTextLeft` - Donada una posició i una cadena de text escriurà el text per l'esquerra de la posició donada.
- `RenderTextRight` - Donada una posició i una cadena de text escriurà el text per la dreta de la posició donada.
- `RenderTextCenter` - Donada una posició i una cadena de text escriurà el text centrat a la posició donada.

Si el professor volgués utilitzar una font nova es pot modificar el fitxer `fonts-config.xml` per afegir noves fonts. L'estructura d'un node 'Font' a aquest XML és la següent:

```
<FONT File='XXX' Id='XXXX' pixel_offsetX='X' pixel_offsetY='X' font_colors='X'
alpha='X'/>
```

- `File` - Nom del fitxer de font, un fitxer .bmp que es trobi a la carpeta Fonts.
- `Id` - Identificador de la font per més tard poder utilitzar-lo.

- pixel_offsetX - Valor en píxels del desplaçament entre caràcter i caràcter de la font.
- pixel_offsetY - Valor en píxels del desplaçament entre línies de color de la font.
- font_colors - Quantitat de colors diferents que hi ha al fitxer de la font, coincidirà amb el nombre de línies de la font.
- alpha - Ens indica si es vol renderitzar amb alpha o no. 0 - Es renderitzarà la font sense tenir en compte un color alpha, 1 - Es renderitzarà la font agafant com a color alpha el píxel [0,0] de la imatge.

La [figura 3.2] ajudarà a entendre millor els camps pixel_offsetX i pixel_offsetY :



Figura 3.2:
(blau) - pixel_offsetX
(verd) - pixel_offsetY

3.4 Mòdul de gestió d'entrada

El mòdul de gestió d'entrada és l'encarregat de gestionar tot el relacionat amb les accions de l'usuari ja sigui amb el teclat o el ratolí. Controla si una tecla ha estat apretada per l'usuari, en quina posició es troba actualment el ratolí de la pantalla i quins botons del ratolí s'estan apretant al moment.

Consisteix en un mapa on es guarda la informació de l'estat del teclat i del ratolí per frame. D'aquesta manera també es pot controlar, si una tecla s'està

mantenint apretada. Tota aquesta informació ens serà proporcionada gràcies a les funcions que control·len l'entrada de hardware de la llibreria SDL. Inclús la llibreria SDL, proporciona ja les estructures per guardar tota aquesta informació.

3.5 Mòdul de gestió del temps

El mòdul de gestió del temps és l'encarregat de gestionar el pas del temps i els temporitzadors. Al joc necessitarem saber exactament quants milisegons han passat d'un frame al següent frame, ja que amb aquest valor calcularem l'increment de tota la lògica del joc. Ja siguin els enemics, ja siguin les torres, absolutament tot el joc, funcionarà en base al temps entre frame i frame. Això és així ja que en una aplicació a temps real, depenent el Hardware en que s'executi, anirà a més frames per segon o menys, i si no controlem la lògica mitjançant l'increment de temps entre frame i frame i ho fem per cada frame, veuríem que a un ordinador el joc va a la velocitat correcta i en altres més potents el joc va massa ràpid. D'aquesta manera ens assegurem de que el joc anirà sempre a la mateixa velocitat en qualsevol tipus de hardware.

Aquest mòdul també gestionarà una part molt important del joc, els temporitzadors. Els temporitzadors són disparadors que al passar un cert temps executen una funció donada (CallBack). Això és molt útil per si volem que diversos events passin al cap d'un temps concret. En el joc seran utilitzats per exemple en la sortida d'enemics, es configurarà un temporitzador que cada cert temps en mil·lisegons dispari la sortida d'un enemic.

La informació sobre el temps transcorregut, es retreu amb funcions específiques de la llibreria SDL, que ha facilitat molt més aquesta tasca. També SDL proporciona funcions per a controlar els temporitzadors fet que ha fet més simple la implementació d'aquest mòdul.

3.6 Mòdul de gestió de configuracions XML

El mòdul de gestió de configuracions XML és l'encarregat de gestionar tot l'accés a els fitxers XML. Utilitzant la llibreria pugiXML aquest mòdul aporta la lectura dels diferents XML així com la notificació d'errors d'aquests.

Capítol 4

Implementació del joc

4.1 Vista global

El videojoc finalment implementat consisteix en un joc simple de defensa de torres. El jugador, en el nostre cas un alumne, comença amb una quantitat configurable de vides, de recursos i nombre de pauses màximes configurable per l'professor. El joc consisteix en que l'alumne, mitjançant la col·locació de torres, ha d'evitar que els enemics, que segueixen un camí definit pel professorat, arribin al final d'aquest. Si els enemics arriben al final del camí el jugador perdrà una vida, si les perd totes acabarà la partida.

Està dividit en diverses rondes configurades a decisió del professor. Cadascuna d'aquestes rondes conté varis enemics que aniran sortint sense parar, mentre al jugador no se li acabin les vides i sempre que els enemics d'una d'elles arribin tots al final o bé hagin mort, s'anvarà de ronda .

Si el jugador aconsegueix acabar totes les rondes configurades tenint una vida o més, haurà superat el joc.

La interfície del videojoc permetrà al jugador pausar la ronda en curs, augmentant el comptador de pauses realitzades. Mentre la partida estigui en pausa

L'alumne no podrà en cap cas, construir noves torres, però sí que podrà visualitzar les estadístiques de mal de cada torre. S'ha de tenir en compte que el professor podrà configurar el màxim nombre de pauses que es podran realitzar a cada partida. Per tant és un altre factor de dificultat afegit, ja que el jugador haurà d'optimitzar molt bé la utilització de totes aquestes pauses.

El jugador, també podrà reiniciar una ronda de la partida ja sigui perquè ha col·locat malament alguna torre o bé per qualsevol altre motiu. Reiniciar la ronda suposa eliminar les torres que ha afegit noves en aquesta ronda i retornar els recursos invertits en aquestes.

Al executar el joc el primer que es trobarà el jugador és el menú inicial [figura 4.1] on apretant qualsevol tecla entrarà dins la partida.



Figura 4.1: Menú del joc

Un cop iniciat el joc [figura 4.2], es carregarà tota la configuració sobre els

elements del joc, les torres, les rondes i els enemics a partir de fitxers XML. En aquest punt es presentarà al jugador la interfície de joc, que consta d'una quadrícula amb uns camins marcats que faran els enemics, els botons per construir torres, botons per controlar el flux del joc, com per exemple el botó d'iniciar ronda i a part es podrà visualitzar, a dalt a la dreta, la informació necessària pel jugador, els recursos disponibles, la ronda actual, el nombre de pauses, les vides restants i el temps transcorregut.

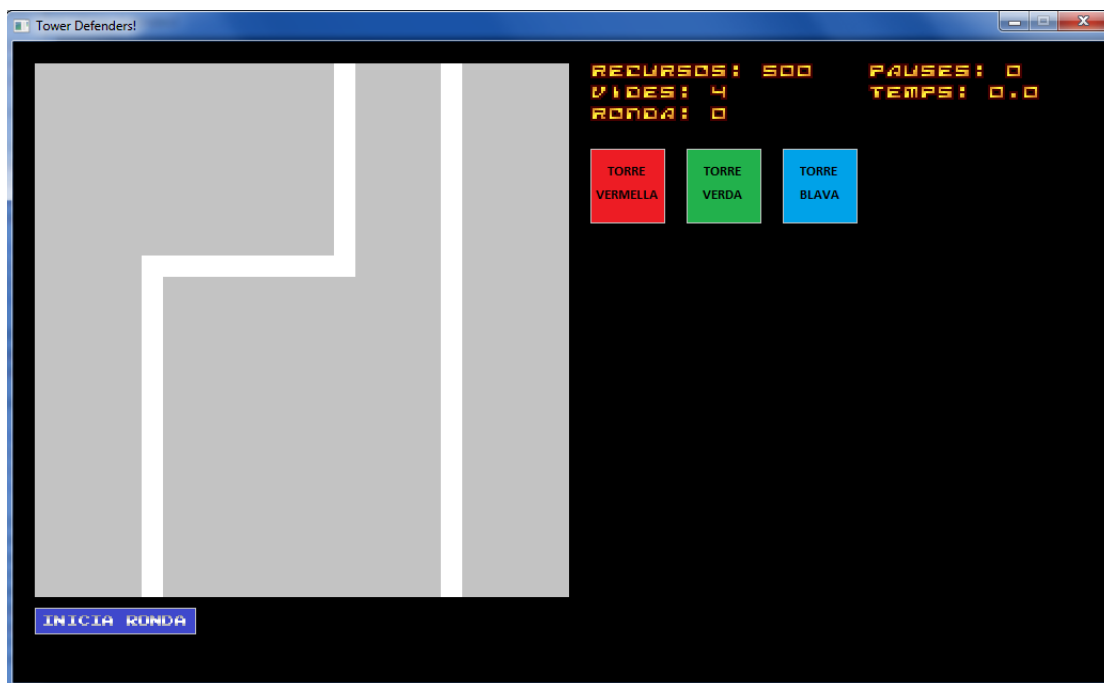


Figura 4.2: Interfície del joc inicialment

En el joc, a cada ronda [figura 4.3], hi haurà enemics de tres colors diferents colors, vermell, verd i blau. Tot i així es permetrà la configuració total de les característiques d'aquests enemics, podent crear enemics del mateix color que vagin més ràpid o més lents o bé que tinguin més vida o menys entre d'altres. El color només ajudarà a que l'alumne tingui una ràpida orientació a l'hora de seleccionar la torre del mateix color de l'enemic ja que serà la que li causarà el màxim mal.

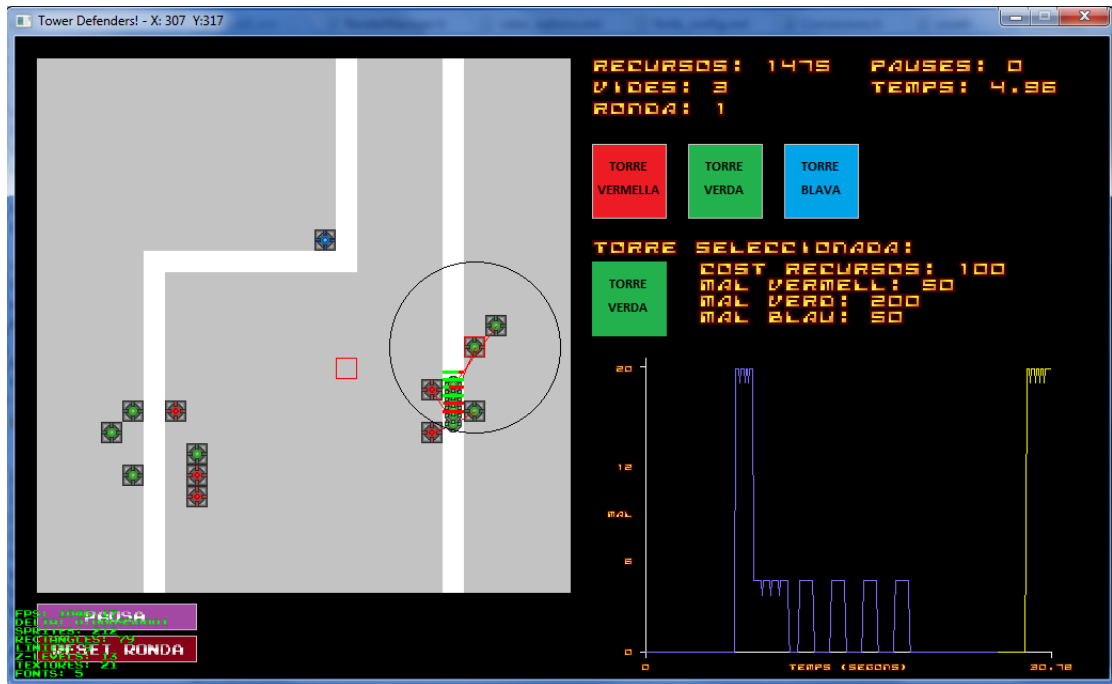


Figura 4.3: Interfície del joc en una ronda

4.2 Lògica i Motor d'Estats

Una part molt important del joc recau en el disseny de la màquina d'estats relacionada amb aquest. Per entendre clarament el disseny final implementat descriurem la utilitat de cada estat després de la [figura 4.4] que mostra els diferents estats amb les seves condicions de transició.

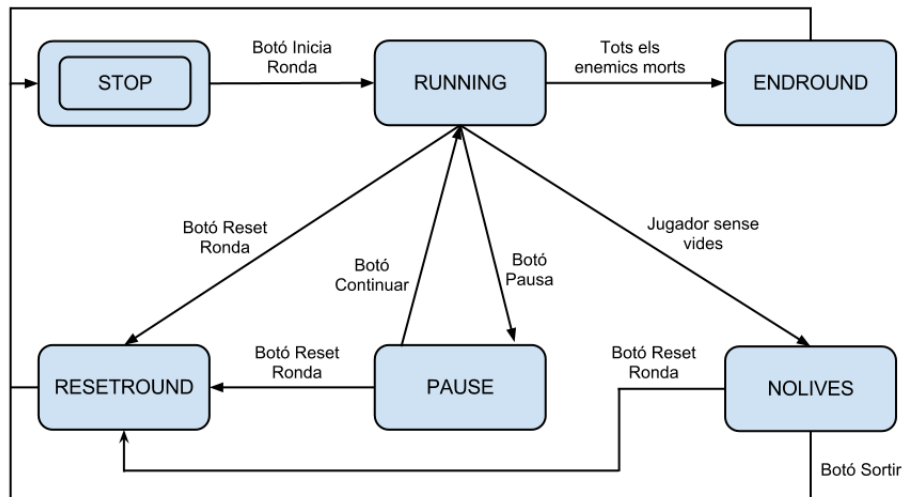


Figura 4.4: Diagrama d'Estats del joc

- **STOP** - Aquest, és l'estat inicial on cada ronda comença, s'encarrega de carregar el nivell d'inici. En aquest estat no surten enemics, es poden construir torres i també visualitzar les estadístiques d'aquestes.
- **RUNNING** - Aquest estat és l'estat on es produeix el moviment d'enemics. En aquest estat es poden construir torres i visualitzar les estadístiques d'aquestes.
- **PAUSE** - Aquest és l'estat en que el joc va quan s'apreta el botó de pausa. En aquest estat es comptabilitzarà una pausa, s'aturarà el moviment d'enemics, no permetrà construir noves torres però sí que permetrà visualitzar les estadístiques de cadascuna d'elles. En el cas de que el jugador hagi esgotat el nombre de pauses disponibles aquest estat tornarà directament a l'estat **RUNNING**.
- **ENDROUND** - Aquest és l'estat en que el joc anirà quan una ronda hagi

acabat, és a dir, quan o bé estiguin tots els enemics morts o hagin arribat al final del camí els que estiguin vius. En aquest estat es realitzarà la comptabilitat dels recursos i carregarà la següent ronda.

- **RESETROUND** - Aquest és l'estat en que el joc anirà quan s'apreta el botó de 'Reset Ronda'. S'encarregarà d'esborrar totes les torres que s'hagin construït en la ronda actual, resetejar els recursos que el jugador tenia al iniciar la ronda actual.
- **NOLIVES** - Aquest és l'estat en que el joc anirà quan s'acabin les vides del jugador. Permetrà visualitzar les estadístiques de cada torre, reiniciar l'última ronda si es desitja o bé sortir al menú principal.

Amb aquest disseny, el jugador inicialment es trobarà a l'estat STOP on podrà invertir els recursos per construir torres per intentar eliminar el màxim possible d'enemics o bé directament podrà iniciar la ronda perquè comencin a sortir aquests. Un cop iniciada la ronda, mitjançant el botó de 'Iniciar Ronda', es canviarà a l'estat RUNNING on començaran a sortir els enemics i el jugador podrà continuar construint noves torres per intentar eliminar els enemics. Si la vida d'un enemic arriba a 0 aquest morirà i donarà una quantitat configurable de recursos al jugador, d'aquesta manera el jugador podrà seguir construint noves torres a mesura que vagi eliminant enemics. Quan una ronda acabi, el joc passarà per l'estat ENDROUND, aquest s'encarregarà de comptabilitzar els recursos finals i carregar la següent ronda, després el joc tornarà a l'estat STOP, en aquest moment el jugador podrà mirar les estadístiques de les seves torres i veure quines conclusions pot treure sobre la seva estratègia. De nou, per iniciar la següent ronda, el jugador utilitzarà el botó de 'Iniciar Ronda' per passar a l'estat de RUNNING i que comencin a sortir els enemics de la següent ronda. Així successivament fins acabar les rondes configurades.

4.3 Elements Configurables

El joc consta de varis elements plenament configurables per el professorat. Aquesta configuració es realitza bàsicament mitjançant fitxers XML. D'aquesta manera es proporciona un grau de llibertat molt gran per que el professorat pugui configurar el joc amb el nivell de dificultat que trobi adient.

4.3.1 Nivell

Aquest element és l'encarregat de controlar els valors globals de configuració d'una partida a la partida. Aquests valors globals són els recursos inicials que rep el jugador a l'inici de la partida, les vides inicials que rep el jugador a l'inici de la partida, la quantitat de rondes de la que consta el joc i el nombre màxim de pauses que el jugador podrà realitzar.

Per configurar aquests paràmetres del videojoc editarem en el fitxer levels.xml els camps següents:

```
<LEVEL_INFO file_path='X' recursos='X' vides='X' rondes='X' max_pauses='X'/>
```

- file_path - Ruta al fitxer de configuració d'un nivell.
- recursos - Determina els recursos inicials que rebrà el jugador al començar la primera ronda.
- vides - Determina el nombre total de vides que rebrà el jugador al començar la primera ronda.
- rondes - Determina el màxim de rondes que tindrà el nivell.
- max_pauses - Determina el màxim nombre de pauses que el jugador podrà realitzar.

4.3.2 Camins

Els camins al videojoc són el recorregut que faran els enemics desde la seva creació fins la seva destrucció. Es poden crear la quantitat de camins que el professor desitji configurant-los al fitxer level1.xml. El nom del node al fitxer XML per crear nous camins amb els paràmetres corresponents és el següent:

```
<PUNT id_cami="X"pos_X="X"pos_Y="X"/>
```

- id_cami - Determina l'identificador de camí.
- pos_X - Determina la posició del punt en l'eix Y, d'un punt al camí.
- pos_Y - Determina la posició del punt en l'eix X, d'un punt al camí.

Cada camí ve indentificat pel camp id_cami , per tant el primer punt amb un identificador concret serà l'inici del camí amb l'identificador corresponent. Els següents punts amb aquest identificador seran els següents punts per on els enemics aniran passant. L'últim punt configurat al fitxer amb un identificador concret, serà el final del camí. Si arriben els enemics a aquest últim punt és quan es descomptarà una vida del jugador.

Per visualitzar els camins i punts carregats, si es prem el botó F2 durant una partida es visualitzaran, d'aquesta manera pot ajudar a veure si els camins estan ben configurats tal com el professorat vulgui. Els punts de color verd, representen els punts inicials dels camins. Per ajudar a configurar correctament les coordenades dels punts dels camins, a la barra superior de la pantalla del joc es mostrarà la coordenada on es troba situat el ratolí dins la finestra del joc, amb la tecla F1 es pot activar o desactivar la visualització de les coordenades.

En la [figura 4.5] podem observar com es visualitzen els camins dins el joc:

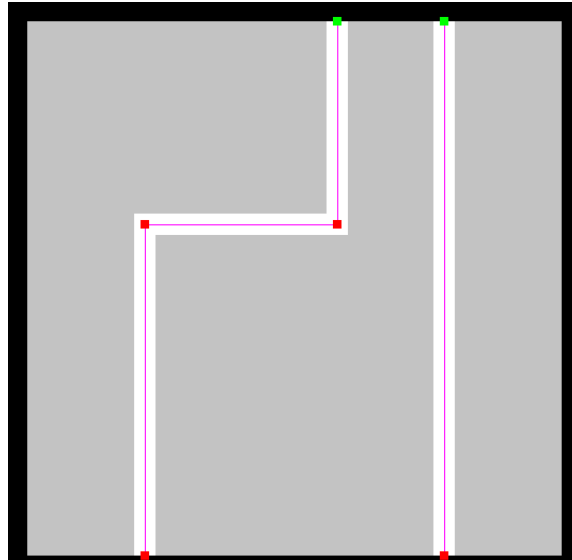


Figura 4.5: Mostreig dels diferents camins

Com es pot comprovar, hi ha una total llibertat a l'hora de crear camins per després assignar-los als enemics a elecció del professorat. Veurem un exemple per il·lustrar més concretament com es configura la creació de, per exemple, els dos camins de la [figura 4.5]:

```
<PUNT id_cami="0"pos_X="310"pos_Y="20"/>
<PUNT id_cami="0"pos_X="310"pos_Y="210"/>
<PUNT id_cami="0"pos_X="130"pos_Y="210"/>
<PUNT id_cami="0"pos_X="130"pos_Y="520"/>
<PUNT id_cami="1"pos_X="410"pos_Y="20"/>
<PUNT id_cami="1"pos_X="410"pos_Y="520"/>
```

En aquest cas, el punt inicial del camí amb identificador 0 seria a la coordenada (310,20), de color verd, el següent punt cap a on viatjaran els enemics seria el (310,210) i així successivament fins arribar al punt final, amb coordenada (130,520). Amb el camí amb identificador 1 passa exactament el mateix però només hi ha dos punts, per tant el punt inicial es el (410,20), en verd, i el final (410,520).

4.3.3 Enemies

Els enemics són els elements del joc que recorreran els camins de principi a fi. Són creats a partir de temporitzadors segons el seu temps de sortida un darrere l'altre a l'inici de un camí. Un cop hagin estat creats viatjaran pel camí amb una certa velocitat configurada fins arribar al final del camí. Mentre realitzen aquest recorregut si reben l'impacte del raig d'una torre es decrementarà la seva vida, que es pot saber fàcilment visualitzant la barra de color que surt sobre de cada enemic. Aquesta barra [figura 4.6] serà plena de color verd quan l'enemic tingui tots els punts de vida. A mesura que vagi rebent mal de les torres aquesta barra anirà tornant-se vermella percentualment fins que arribi a zero punts de vida, moment en que s'eliminarà l'enemic i es recompensarà al jugador amb els recursos configurats per la destrucció d'aquell enemic.

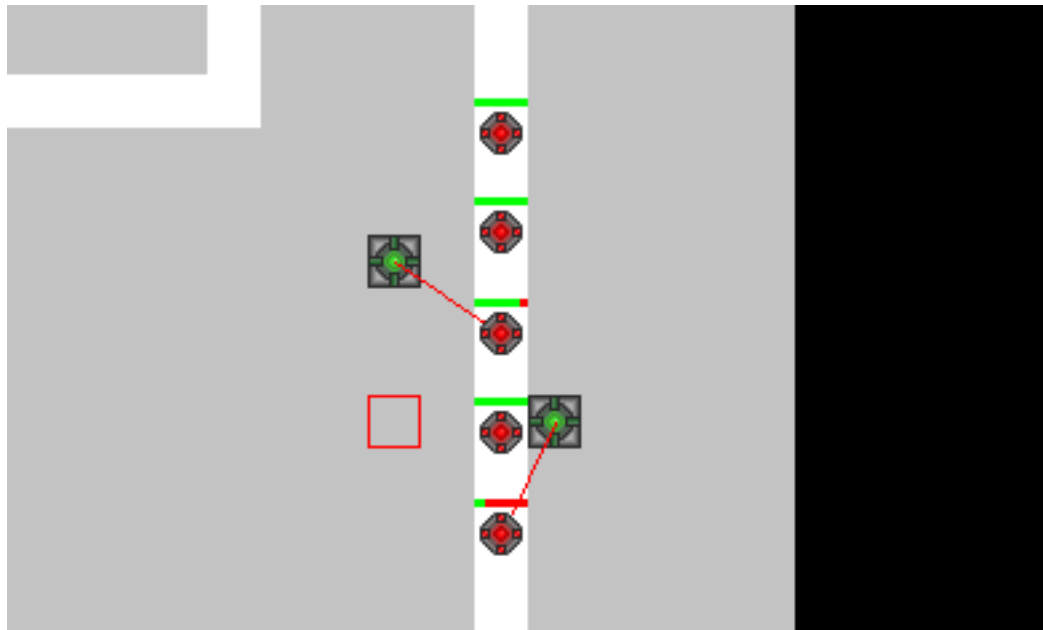


Figura 4.6: Barres de vida dels enemics

Per facilitar la configuració de rondes diferents s'ha optat per permetre la creació de tipus d'enemics de manera que es puguin utilitzar en diferents rondes, sense

haver de definir els seus paràmetres de nou. D'aquesta manera es pot definir un tipus d'enemic concret i fer-lo sortir en diferents rondes molt fàcilment.

Per tal que els professors puguin crear nous tipus d'enemics, per més tard afegir-los a les rondes que creguin necessàries, s'hauràn d'afegir nodes 'ENEMIC_TIPUS' al fitxer level1.xml.

Els paràmtres d'un nou tipus d'enemic són els següents:

```
<ENEMIC_TIPUS id_enemic='X' id_textura='X' velocitat='X'  
vida='X' temps_respawn='X' color='X' recursos_al_morir='X'/>
```

- id_enemic - Determina l'identificador únic d'un tipus d'enemic.
- id_textura - Determina l'identificador d'una textura, aquesta textura servirà a l'hora de pintar l'enemic per pantalla.
- velocitat - Determina la velocitat de l'enemic al llarg del camí, en píxels per segon.
- vida - Determina la quantitat de punts de vida que té un enemic quan es crea.
- temps_respawn - Determina el temps en milisegons entre la sortida d'enemics d'aquest tipus.
- color - Determina el color de l'enemic, només pot ser 'RED', 'GREEN' o 'BLUE'.
- recursos_al_morir - Determina la quantitat de recursos que obtindrà el jugador quan un enemic d'aquest tipus mori (arribi la seva vida a zero).

Un cop s'hagin definit els tipus d'enemics que es desitji per posar-los en una ronda concreta només caldrà afegir al fitxer `level1.xml` nodes del tipus 'ENEMICS' indicant els següents paràmetres:

- `id_enemic_tipus` - Determinarà l'identificador del tipus d'enemic a utilitzar (prèviament definit al fitxer XML de configuració).
- `id_cami` - Determinarà l'identificador del camí per on els enemics viatjaran (prèviament definit al fitxer XML de configuració).
- `quantitat` - Determinarà la quantitat màxima d'enemics que s'aniran creant .
- `ronda` - Determinarà en quina ronda s'iniciarà la sortida d'aquests enemics.

Per dificultar la tasca als alumnes es podrà afegir més d'un tipus d'enemic a la mateixa ronda i al mateix camí. S'ha de tenir en compte, però, que sortiran per ordre d'inserció en el fitxer de configuració XML `levels1.xml`. D'aquesta manera per un camí podem configurar que surtin per exemple 5 enemics del tipus 1 i seguidament per el mateix camí que surtin 10 enemics del tipus 3. Tot això fa que el professorat tingui control total sobre la configuració dels enemics que surten per cada ronda i per cada camí.

El node 'ENEMICS' té un paràmetre molt important, la ronda. Aquest camp és el que s'utilitzarà per determinar en quina ronda sortirà aquest grup d'enemics . D'aquesta manera es pot separar la partida en diferents rondes a gust del professor.

4.3.4 Torres

Les torres són l'únic element del joc que els jugadors podran construir gastant els seus recursos per intentar que els enemics no arribin al final del camí. Aquestes es podran situar sobre de la quadrícula, però només en quadrats on o bé no hi hagi cap altra torre construïda o bé el quadrant formi part del camí dels enemics. Per

fer més difícil la tasca de posicionament de les torres a la quadrícula per part dels alumnes, aquestes només es podran construir si la partida està en curs, és a dir, si està en pausa, no es podran construir torres. Un cop construïdes [figura 4.7], no es podran eliminar excepte si l'alumne prem el botó de reiniciar ronda.

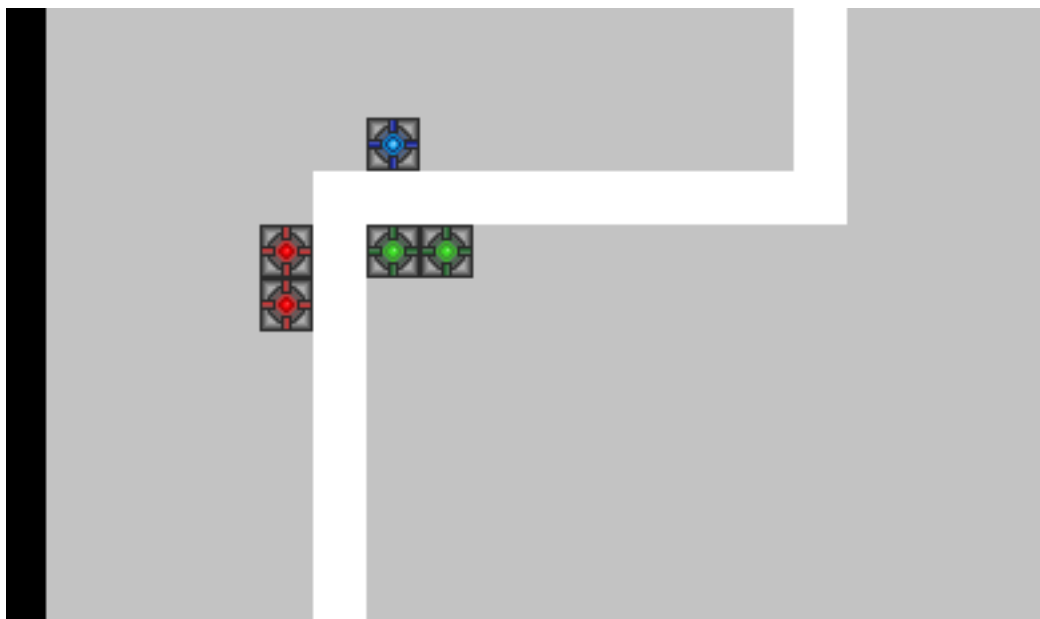


Figura 4.7: Torres construïdes.

Hi ha configurades per defecte fins a tres tipus diferents de torres. Aquestes torres tindran valors de dany als enemics ben diferents i a més configurables. Les torres configurades per defecte són de color vermell, verd i blau, amb valors de dany més elevat als enemics del mateix color que la torre. Això s'ha decidit així per mantenir una relació coherent per que els alumnes entenguin millor que si una torre és de color vermell farà més mal als enemics de color vermell, i així amb tots els colors. No obstant res prohibeix al professor configurar-ho totalment diferent.

Les característiques de les torres són totalment configurables per part del professorat. Els professors podran canviar els següents paràmetres de les torres editant el fitxer de configuració `level1.xml` al node 'TORRE':

```
<TORRE nom='XXX' preu='X' grafic_boto='X' grafic_torre='X' distancia_pixels='X'  
mal_vermell_per_segons='X' mal_verd_per_segons='X' mal_blau_per_segons='X'/>
```

- nom - Determina el nom de la torre. Per exemple 'Torre Vermella'.
- preu - Determina el cost en recursos que se li restaran al jugador quan construeixi la torre.
- grafic_boto - Identificador de la textura a utilitzar per dibuixar el gràfic del botó de la torre.
- grafic_torre - Identificador de la textura a utilitzar per dibuixar el gràfic de la torre a la quadrícula.
- distancia_pixels - Determina el radi d'acció de la torre. És la distància màxima en la que la torre dispararà.
- mal_vermell_per_segons - Determina el mal per segons als enemics de color vermell.
- mal_verd_per_segons - Determina el mal per segons als enemics de color verd.
- mal_blau_per_segons - Determina el mal per segons als enemics de color blau.

Per construir les diferents torres l'alumne simplement haurà de prémer el botó de la torre en concret que vulgui construir amb el botó esquerre del ratolí. Un cop fet això podrà construir-la, sempre que tingui recursos, dins la quadrícula. En el cas de que l'alumne vulgui cancel·lar la construcció d'una torre, podrà desseleccionar-la si prem el botó dret del ratolí.

Quan l'alumne seleccioni una torre o bé quan vagi a construir-ne una de nova, obtindrà una ajuda visual [figura 4.8] de l'abast d'aquesta torre. Això li permetrà saber amb exactitud fins a on arribaran els trets d'aquella torre, sigui on sigui que l'estigui situant.

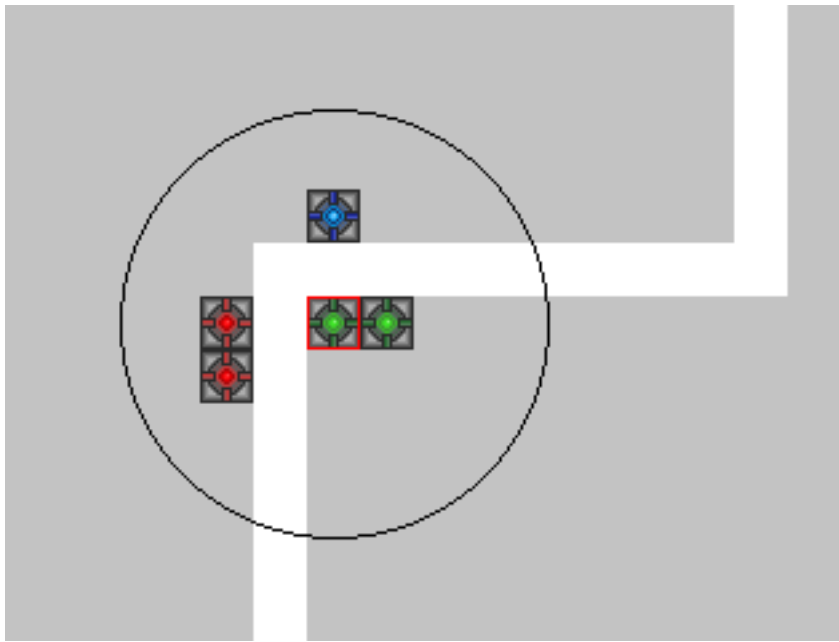


Figura 4.8: Abast d'una torre seleccionada.

4.4 Estadístiques

L'estadística sempre ha estat molt important a l'hora de representar, analitzar i interpretar les dades. Per tant en un joc com el realitzat és sumament important que els alumnes puguin observar com es desenvolupa estadísticament la seva estratègia de joc en temps real.

Per fer-ho s'ha implementat un visualitzador d'estadístiques per a cada torre on l'alumne podrà observar el mal que fa aquella torre en funció del temps. El mal que realitza una torre anirà condicionat per les seves característiques, ja sigui el mal per segon base de la torre, el seu abast, la seva situació a la quadrícula i el

color de l'enemic.

L'alumne només caldrà que en qualsevol estat del joc seleccioni amb el botó esquerra qualsevol de les torres que hi té construïdes. Al seleccionar una torre, es mostrarà a la part inferior dreta del joc una gràfica amb el mal per segon acumulat que aquesta torre ha estat fent desde la seva creació.

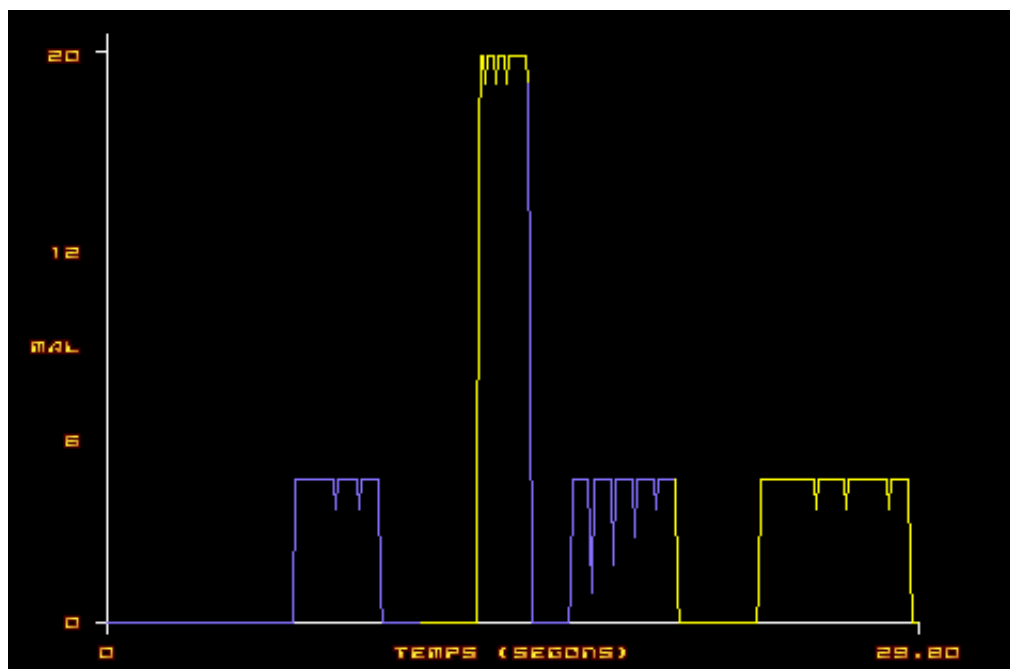


Figura 4.9: Estadístiques.

Com es pot observar a la [figura 4.9], aquesta gràfica ve donada en funció del temps, i ens proporciona informació sobre el mal per segon que està fent la torre seleccionada. Si observem la gràfica veiem que la línia pren diferents colors, per tal que sigui més senzill diferenciar les rondes. És a dir que el color de la línia anirà variant segons canvi la ronda. D'aquesta manera l'alumne podrà discernir ràpidament en quina ronda ha estat o no efectiva la torre seleccionada i per tant també si la seva estratègia de col·locació de la torre es la correcta.

Aquesta és una eina clau a l'hora d'ensenyar als alumnes, ja que amb aquesta

gràfica el professor podrà realment explicar el comportament d'una torre envers uns enemics i els alumnes podran avaluar si la seva estratègia és correcta, millorable o incorrecta.

Capítol 5

Conclusions

En aquest capítol es fa un repàs als objectius assolits, i les possibles millores en un treball futur així com les valoracions d'un professor que ha pogut provar l'aplicació.

5.1 Compliment dels Objectius

Els objectius generals del projecte s'han complert, més concretament el projecte s'ha dividit en 4 objectius clars:

- Que estigui dotat d'un entorn gràfic senzill que inclogui un joc de defensa de torres amb els elements principals d'aquest tipus de joc.

S'ha assolit aquest objectiu proporcionant un joc gràficament senzill però funcional, on els jugadors, en aquest cas, els alumnes, trobin la informació que cal sobre la partida i puguin interactuar amb el joc de manera senzilla i intuïtiva. El joc finalment implementat conté els elements principals d'aquest tipus de joc, torres, enemics, camins i botons.

- Que el joc estigui dissenyat de forma neutra sense una ambientació que pugui provocar una discriminació entre els alumnes.

S'ha assolit la neutralitat en l'ambientació gràfica. Tot i així gràcies a la plataforma, es poden canviar els gràfics del joc molt fàcilment per part del professorat.

- Que els elements que determinen el nivell de dificultat del joc siguin configurables pel professorat que l'ha de fer servir (camins, enemics, torretes..) per poder-lo adequar a diferents activitats i nivells educatius.

Aquest objectiu ha quedat assolit gràcies a un total control sobre les diferents característiques configurables mitjançant fitxers XML de tots els elements que intervenen en el joc. D'aquesta manera queda a total control del professor configurar el joc per adequar-lo al nivell de dificultat que vol aplicar al seus alumnes, en aquest cas els jugadors.

- Que contingui elements que permetin a l'usuari, en aquest cas, l'alumne avaluar el grau de desenvolupament del seu joc.

Aquest objectiu s'ha assolit proporcionant les estadístiques en temps real del mal que fan les torres. Aquesta és la eina essencial que servirà a l'alumne per observar si la seva estratègia ha estat la correcta.

5.2 Ampliacions o millores

El joc finalment implementat només disposa de tres torres configurables que no més disparen a un sol enemic. Una millora que podria afegir és que s'hi poguessin configurar torres que disparin a més d'un enemic a l'hora, això faria que hi hagués molta més varietat alhora de que els alumnes pensessin la situació d'aquestes torres i per tant la seva estratègia també.

Un altre aspecte, a nivell de configuració, que es podria afegir seria la inclusió d'actualitzacions configurables de les torres, és a dir, que donada una torre que ja està construïda, es pogués actualitzar aquesta perquè fes més mal, o bé augmentar el seu abast. Això permetria encara més llibertat a l'hora de triar una estratègia però segurament dificultaria força el balanceig de la dificultat del joc, ja que els

professors tindrien moltíssimes més variables a tenir en compte a l'hora de configurar una partida.

Una ampliació que seria molt útil, seria la de crear un editor visual que serveixi per no haver de manipular els fitxers XML directament. Seria molt útil per als professors per fer les seves configuracions.

Com tots els jocs, un aspecte que no s'ha afegit però que augmentaria molt més la immersió del joc seria la introducció de so i música. Això es podria aconseguir amb un nou mòdul a l'entorn i després fer-lo servir al joc.

5.3 Valoracions d'un professor

El format de joc Tower Defense obre noves possibilitats per acostar a les aules els usos reals de les noves tecnologies, incidint en el món dels videojocs. No existeix un consens general que identifiqui la potencialitat dels videojocs com a eina pedagògica. En especial per a les noves generacions d'alumnes que formen part de la gamer generation, alumnes que han conviscut amb els videojocs a la seva vida quotidiana. El Tower Defense desenvolupat en aquest projecte conté opcions que no es poden trobar en les alternatives accessibles per al professorat. El software desenvolupat permetrà als alumnes dissenyar i desenvolupar estratègies per superar el joc a partir de la informació numèrica apareguda a la pantalla, com la que es mostra a la [figura 5.1]:



Figura 5.1: Característiques d'una torre.

També es podran treballar aspectes geomètrics a partir de la disposició de les torretes com es pot observar a la [figura 5.2]:

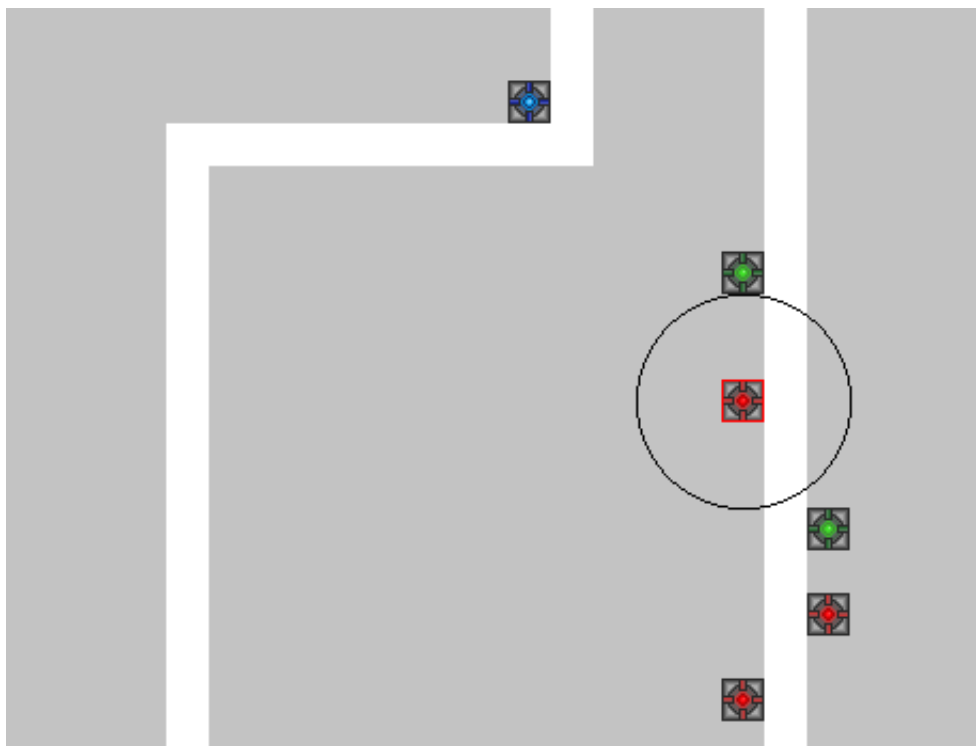


Figura 5.2: Abast d'una torre.

Un dels aspectes més importants, és la possibilitat d'avaluar l'efectivitat de les torretes utilitzades a partir de gràfiques [figura 5.3]. D'aquesta forma els estudiants poden relacionar una realitat concreta (allò que succeeix en el seu joc) amb una informació numèrica que la representa. Aquest fet obre possibilitats per treballar les relacions numèriques que porten al concepte de funció, així com relacionar-les amb la seva gràfica. Per tant, el software desenvolupat permet introduir a l'aula de matemàtiques la possibilitat de discutir sobre una realitat concreta, fàcilment experimentable a partir de la informació gràfica presentada. Una mostra de gràfica que permet induir aquest tipus de discussions és el següent, en el que es pot demanar als alumnes que expliquin l'activitat realitzada per una torreta concreta.

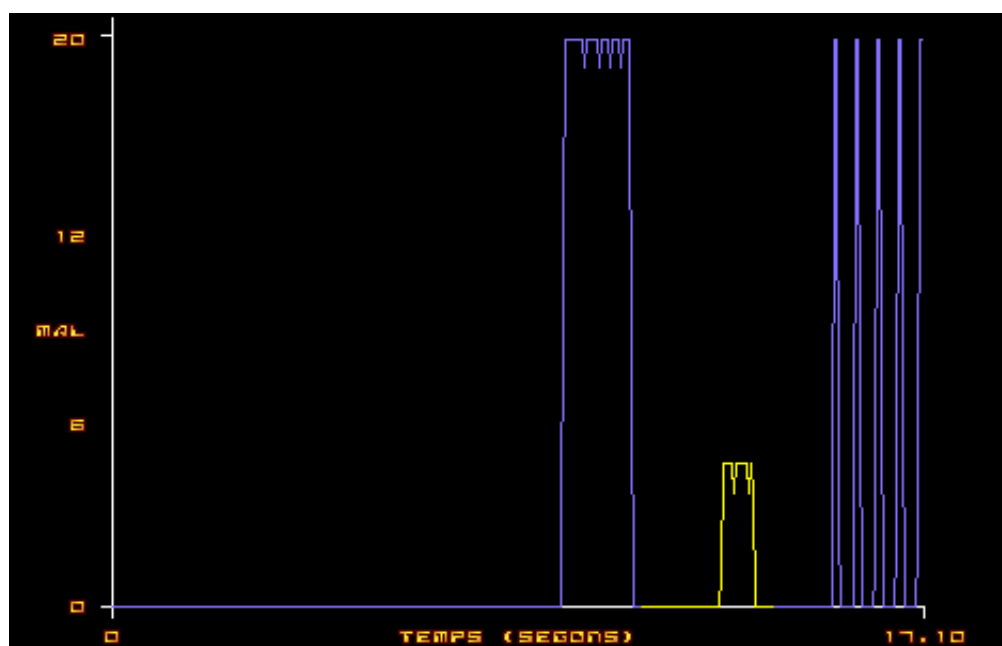


Figura 5.3: Estadístiques d'una torre.

Finalment, la forma en la que s'ha programat aquest Tower Defense permet al professor/investigador ajustar les característiques de dany de les torretes i dels enemics, amb el que es pot ajustar el nivell de joc a diferents nivells educatius o es poden dissenyar escenaris concrets que presentin situacions problemàtiques als

alumnes. Aquest fet, juntament amb els anteriors aspectes comentats, proporciona al software desenvolupat una potència educativa que va més enllà de la que posseeixen els videojocs comercials habituals.

Bibliografia

- [CPP12] cplusplus.com, 2012.
<<http://www.cplusplus.com/>>
- [FOCOS] Focus On SDL (Focus on Game Development), 2002
- [DASGP] Data Structures for Game Programmers, 2002
- [GAMDEV] gamedev.net, 2012.
<<http://www.gamedev.net/page/index.html/>>
- [LIBSDL] libsdl.org, 2012.
<<http://www.libsdl.org/>>
- [WIKSDL] wiki.libsdl.org, 2012.
<<http://wiki.libsdl.org/moin.cgi/>>
- [OGL] opengl.org, 2012.
<<http://www.opengl.org/>>
- [KNGR] kongregate.com, 2012.
<<http://www.kongregate.com/tower-defense-games/>>

Firmat: Daniel Guzmán Cerón
Bellaterra, Setembre de 2012

Resum

L'objectiu principal d'aquest projecte és el d'implementar una plataforma per a l'anàlisi de videojocs educatius del tipus 'Defensa de torres'. Primerament s'ha desenvolupat un entorn/framework en C++ que conté els mòduls principals per gestionar el joc i així ens permet desenvolupar a sobre d'ell un videojoc d'aquest tipus. La plataforma permet un total control sobre les opcions de configuració del joc, així com les característiques de les torres, les característiques dels seus enemics i les rutes dels enemics. En quant a l'anàlisi, la plataforma és capaç de generar estadístiques sobre l'efecte de les torres en temps real, que permeten al jugador en tot moment analitzar la seva estratègia.

Resumen

El objetivo principal de este proyecto es el de implementar una plataforma para el análisis de videojuegos educativos del tipo de 'Defensa de torres'. Primeramente se ha desarrollado un entorno/framework en C++ que contiene los módulos principales para gestionar el juego y así nos permite desarrollar encima de él un videojuego de este tipo. La plataforma permite un control total sobre las opciones de configuración del juego, así como las características de las torres, las características de los enemigos y las rutas de estos. En cuanto al análisis, la plataforma es capaz de generar estadísticas sobre el efecto de las torres en tiempo real, lo que permite al jugador en todo momento, analizar su estrategia.

Abstract

The aim of this project is to implement a platform to analyze 'Tower Defense' kind of educative games. Firstly a framework has been developed in C++ language, that contains all the necessary modules to manage all the aspects in a 'Tower Defense' kind of game. The platform allows total control to configure all the parameters in the game, including the parameters of the towers, the parameters of the enemies and their paths. The platform is capable to generate statistics about the towers in real time, that allows the players to analyze their strategy in real time.